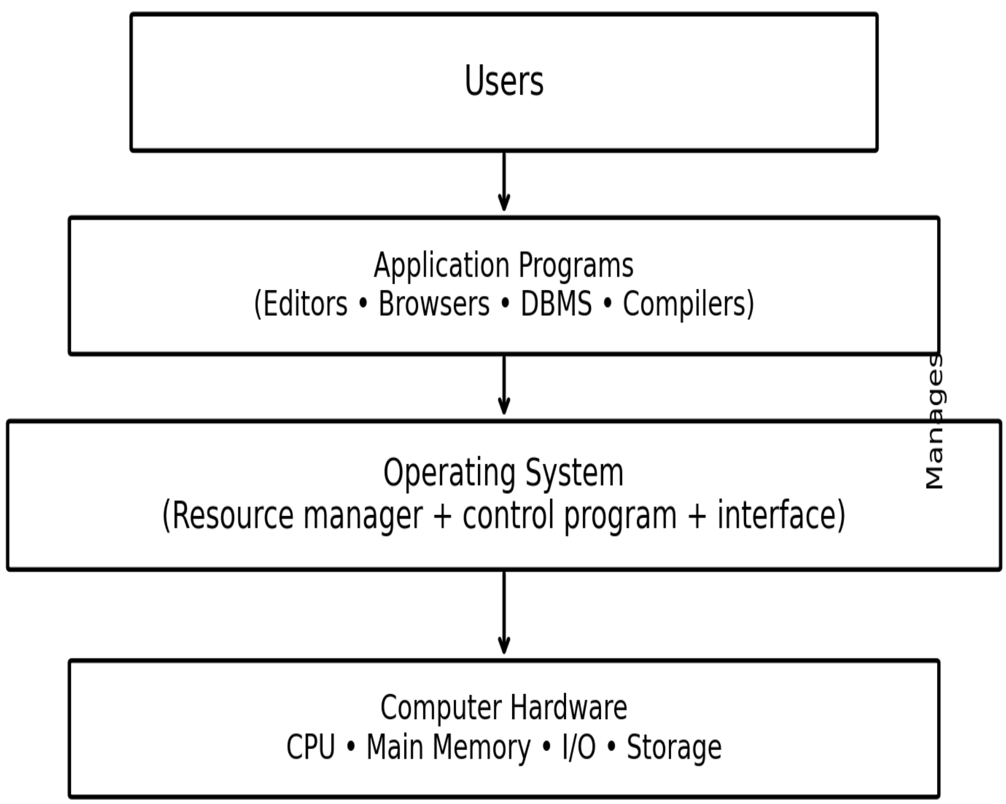
OPERATING SYSTEM

Unit I – Operating System Services and Components

Course Code: 315319 | Semester V | K Scheme

Publishing-ready student notes • syllabus aligned • exam mapped • citation supported

Operating System as the Interface and Resource Manager



Prepared from the prescribed syllabus, the supplied Unit-I notes and Summer 2026 / Winter 2025 question papers.

1. Unit Scope and Learning Outcomes

The prescribed Unit I carries 14 marks and is aligned with CO1. The syllabus expects students to explain the operating-system concept and functions, major OS types, command-line and GUI-based operating systems, OS services and system calls, and the five major OS components.

Syllabus item	Coverage in these notes	Status
1.1 OS: concept, functions	Sections 2–3	Core
1.2 OS types: Batch, Multiprogrammed, Time-shared, Multiprocessor, Distributed, Real-time, Mobile (Android)	Section 4	Core + exam-linked extension
1.3 CLI: DOS, UNIX; GUI: Windows, Linux, macOS	Section 5	Core; classification clarified
1.4 OS services; system calls: concept and types	Sections 6–7	Core
1.5 OS components: Process, Main Memory, File, I/O, Secondary Storage	Section 8	Core

EXAM LINK | Summer 2026: Unit-I opportunities were Q2(a), Q3(a), Q4(a), Q5(a). Winter 2025: Q1(a), Q1(b), Q1(d), Q2(a), Q6(a).

Syllabus basis: 315319 Operating System, K Scheme, Unit I, TLO 1.1–1.4 and CO1.

Learning Outcomes for Unit I

- Describe the concept and functions of an operating system.
- Explain the major types of operating systems and select a suitable type for a given application.
- Differentiate CLI and GUI-based interaction and relate the prescribed examples to each.
- Explain operating-system services and the role and categories of system calls.
- Describe the activities performed by the major operating-system components.

2. Operating System: Concept

An Operating System (OS) is system software that manages computer hardware and software resources and provides services to application programs. It acts as an intermediary between users/applications and the hardware.

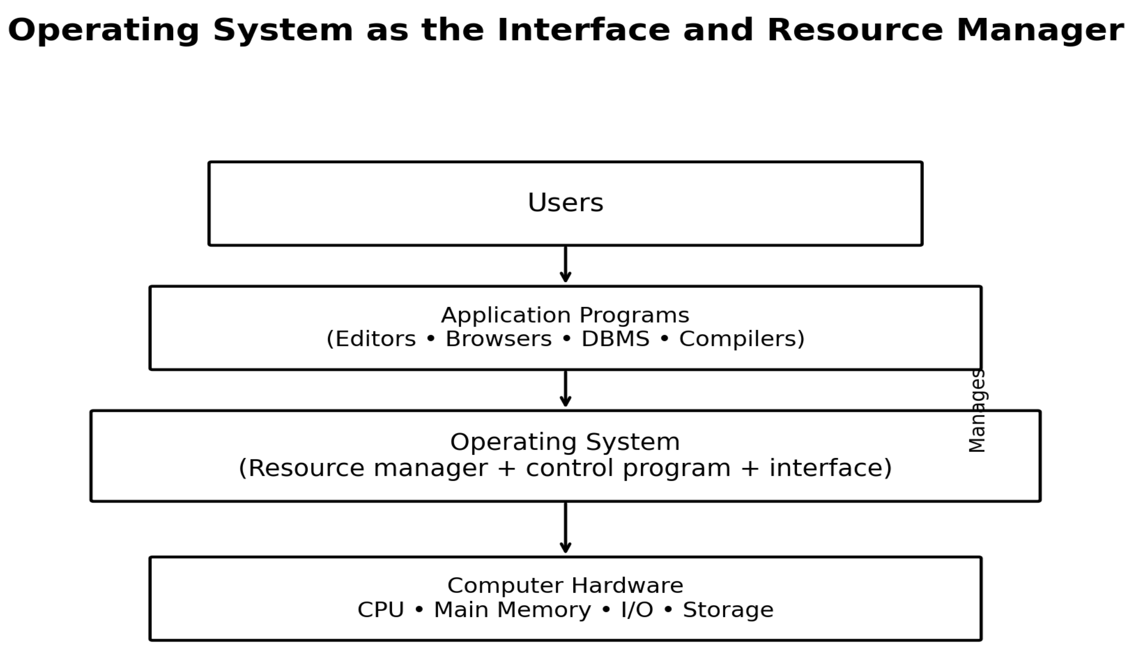


Figure 1. OS as an interface and resource manager.

EXAM LINK | Winter 2025 Q1(d): "What are major differences between Linux and Windows?" and Q6(a): definition of OS and types. These are directly supported by the concept and OS-type sections.

Reference support: Silberschatz, Galvin & Gagne, Operating System Concepts, 10th ed., Ch. 1–2; NPTEL Operating System Fundamentals, Weeks 1–2. cite turn0search0 turn0search1

Computer-system view

Layer	Role	Examples
Users	People or external actors who request computing services	Student, developer, administrator
Application programs	Solve user-level problems using OS services	Editor, browser, compiler, DBMS
Operating system	Allocates resources, controls execution and exposes services	Windows, Linux, macOS, Android
Hardware	Provides physical computing resources	CPU, RAM, I/O devices, storage

EXAM LINK | The supplied notes included a four-part computer-system view. It has been retained but rewritten for clarity and publication quality.

3. Functions of an Operating System

The following functions are the most useful exam-ready formulation for Unit I:

Function	What the OS does	Typical example
Process management	Creates, schedules, suspends/resumes and terminates processes; coordinates process execution.	Running a browser while a file is being downloaded
Main-memory management	Tracks memory use and allocates/deallocates memory to processes.	Assigning RAM to a newly launched program
File management	Creates/deletes files and directories and controls file access.	Opening, saving or deleting a file
I/O and device management	Coordinates input/output devices and device drivers.	Printer, keyboard, disk and network I/O
Secondary-storage management	Manages free space, storage allocation and disk-access requests.	Allocating disk blocks to files
Protection and security	Controls access to resources and authenticates users.	User login and file permissions
Resource allocation	Distributes CPU, memory, storage and I/O resources among competing processes.	CPU scheduling
Accounting	Tracks resource usage by users/processes.	CPU time or storage usage records
Communication	Supports information exchange between processes and systems.	Shared memory or message passing

EXAM LINK | Summer 2026 Q5(a): "Describe any six services of operating system with example." Winter 2025 Q1(a): "State any four operating system services."

The supplied notes listed process, memory, file, device/resource management, accounting, security, communication and booting; the final notes separate OS functions from OS services to match the syllabus and standard textbook treatment.

4. Types of Operating Systems

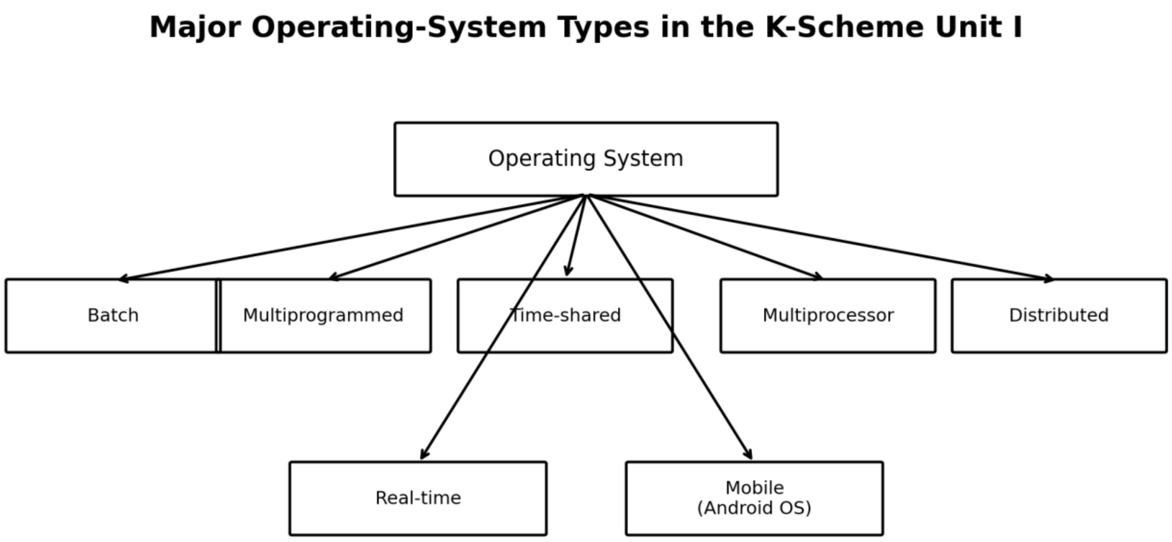


Figure 2. Major OS types required by the K-Scheme syllabus.

4.1 Batch Operating System

A batch OS collects jobs and processes them with little or no interactive user intervention during execution.

- Jobs are grouped into batches and executed sequentially.
- High turnaround time can be a limitation.
- Suitable for repetitive, offline workloads such as large-volume data processing.

EXAM LINK | The supplied notes cover batch processing; no direct batch-only question appeared in the two supplied papers.

4.2 Multiprogrammed Operating System

Multiprogramming keeps several jobs in main memory so that the CPU can switch to another ready job when the current job waits for I/O.

- Main objective: improve CPU utilization and throughput.
- Several jobs share CPU time; only one executes on a single CPU at a time.
- The OS needs memory management and scheduling support.

EXAM LINK | Summer 2026 Q2(a): differentiate multi-programmed and multitasking OS.

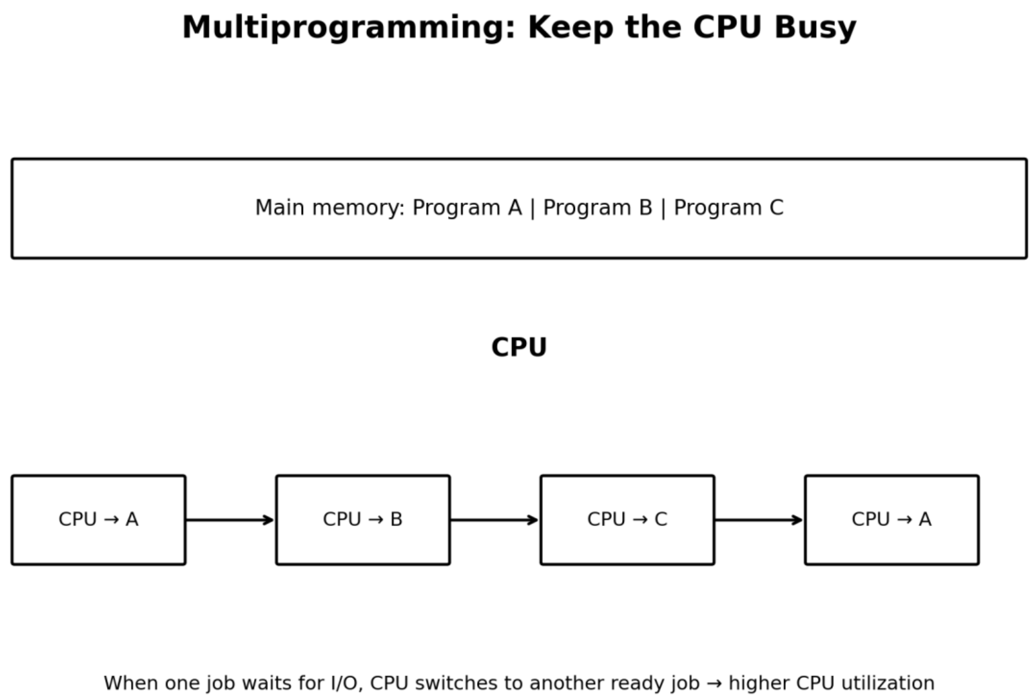


Figure 3. Multiprogramming reduces CPU idle time by switching among ready jobs.

4.3 Multitasking – exam-linked extension

Multitasking allows multiple tasks/processes to make progress concurrently, usually by rapid CPU switching.

- On a single CPU, apparent simultaneous execution is achieved through time slicing/context switching.
- Modern desktop and mobile OSs commonly support multitasking.
- The concept is closely related to multiprogramming but emphasizes concurrent task execution and user responsiveness.

EXAM LINK | Summer 2026 Q2(a) explicitly asks multi-programmed vs multitasking. Although multitasking is not separately named in the Unit-I syllabus list, it is necessary for exam coverage.

Point	Multiprogramming	Multitasking
Primary objective	Increase CPU utilization / throughput	Support concurrent tasks and responsive interaction
Focus	Keeping CPU busy when a job waits	Rapid switching among active tasks
Typical context	Batch and multi-job environments	Interactive desktop/mobile environments
Relation	Foundation for concurrent execution	Commonly implemented using scheduling/time slicing

4.4 Time-Sharing Operating System

Time sharing is an extension of multiprogramming in which CPU time is divided into short time slices (quanta), allowing multiple users/processes to interact with the system.

- Designed for interactive use and quick response.
- Each ready process receives CPU time according to the scheduling policy.
- Common in multi-user systems.

EXAM LINK | The syllabus explicitly includes time-shared OS; no direct question appeared in the two papers, but it is a prescribed topic.

4.5 Multiprocessor Operating System

A multiprocessor OS manages a system containing two or more processors that can cooperate on workloads.

- Processors may share memory and I/O resources.
- Symmetric multiprocessing (SMP) treats processors as peers; asymmetric systems may assign master/slave roles.
- Benefits include higher throughput and improved availability; coordination is more complex.

EXAM LINK | Prescribed by syllabus; not directly asked in the supplied papers.

4.6 Distributed Operating System

A distributed system coordinates multiple autonomous computers connected by a network so that users can access shared resources and services.

- Each node normally has its own local memory and processor.
- Communication occurs through a network.
- Resource sharing and fault tolerance are important benefits.

EXAM LINK | Winter 2025 Q6(a)(i): explain Distributed OS with an example.

4.7 Real-Time Operating System

A real-time OS is designed to respond to events within specified timing constraints.

- Hard real-time systems require critical deadlines to be met.
- Soft real-time systems can tolerate occasional deadline misses.
- Applications include industrial control, avionics and time-critical monitoring.

EXAM LINK | Winter 2025 Q2(a): explain real-time system and list two applications.

4.8 Mobile Operating System – Android

A mobile OS manages the resources, user interaction, applications, connectivity and power constraints of handheld devices. Android is a prescribed example.

- Optimized for touch interaction and mobile hardware.
- Manages application processes, memory, storage, sensors, connectivity and power.

- Provides an application framework and security mechanisms for mobile apps.

EXAM LINK | Winter 2025 Q6(a)(ii): explain Mobile OS with an example.

5. Command-Line and GUI-Based Operating Systems

The syllabus names DOS and UNIX under command-line based systems and Windows, Linux and macOS under GUI-based systems. For technical accuracy, Linux and macOS should not be described as GUI-only: both provide graphical environments as well as command-line shells.

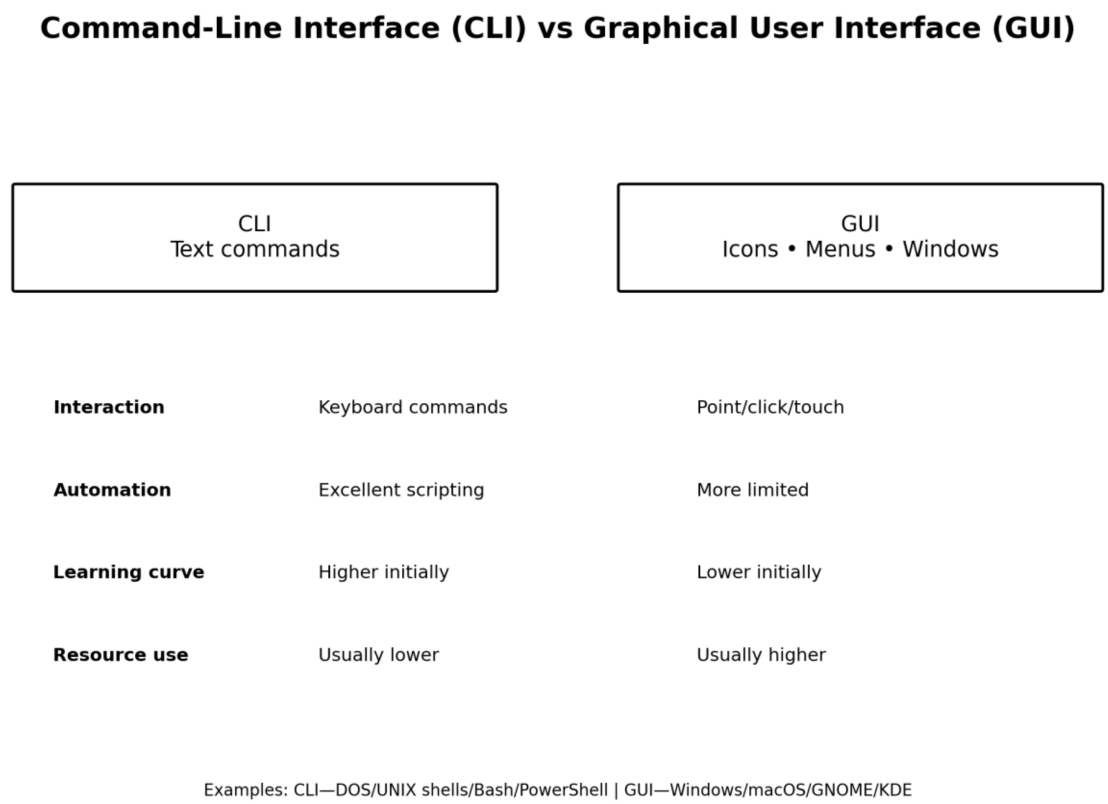


Figure 4. CLI vs GUI – exam-oriented comparison.

Aspect	CLI	GUI
Interaction	Text commands	Windows, icons, menus, pointers/touch
Learning curve	Higher initially; commands must be remembered	Generally easier for beginners
Automation	Excellent for scripting	Possible, but usually less direct for repetitive shell tasks
Resource use	Often lower overhead	Generally higher graphical resource use
Examples in/around syllabus	DOS, UNIX shells, Bash	Windows, macOS, Linux desktop environments

EXAM LINK | Summer 2026 Q2(a) indirectly tests OS interaction concepts through multi-programming vs multitasking; Winter 2025 Q1(d) asks Linux vs Windows. The CLI/GUI distinction is therefore important supporting knowledge.

Linux reference support: Petersen, Linux: The Complete Reference, 6th ed.; official McGraw Hill information confirms coverage of shells, desktops, files and system administration.

6. Operating-System Services

Operating-system services provide a convenient execution environment and help the OS manage the system efficiently.

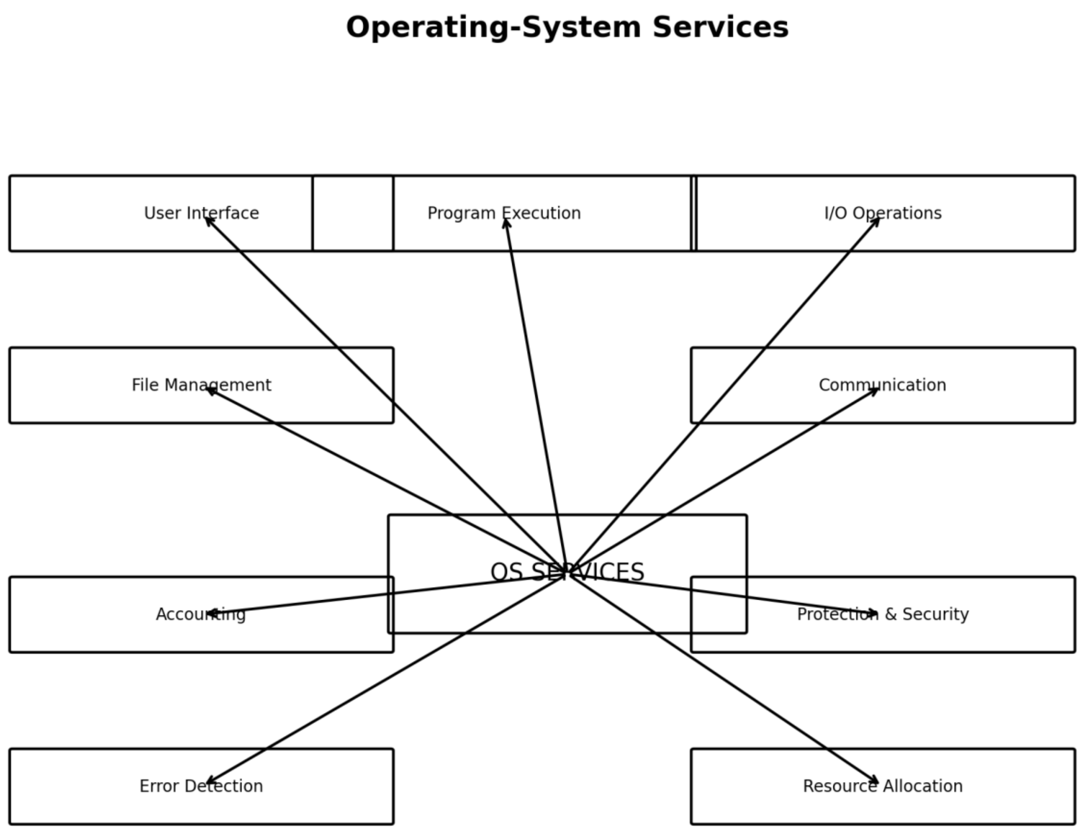


Figure 5. Common operating-system services.

Service	Exam-ready explanation
User interface	Provides CLI, GUI, batch or other interaction mechanisms.
Program execution	Loads a program, starts execution and handles normal/abnormal termination.
I/O operations	Provides controlled access to input/output devices and files.
File-system manipulation	Creates/deletes files and directories, reads/writes data and manages permissions.

Communication	Enables processes to exchange information using shared memory or message passing.
Error detection	Detects and handles hardware/software errors to maintain reliable operation.
Resource allocation	Allocates CPU, memory, storage and I/O resources among competing processes.
Accounting	Records resource usage for users or processes.
Protection and security	Controls access to resources and authenticates users.

EXAM LINK | Summer 2026 Q5(a) asks for six services with examples. Winter 2025 Q1(a) asks for any four services.

Silberschatz, Operating System Concepts, 10th ed., Ch. 2 identifies user interface, program execution, I/O, file-system manipulation, communication, error detection, resource allocation, accounting, and protection/security as common OS services.

7. System Calls

A system call is the programming interface through which a user-level program requests a service from the operating-system kernel. Applications commonly reach system-call functionality through an API or library wrapper rather than invoking the kernel interface directly.

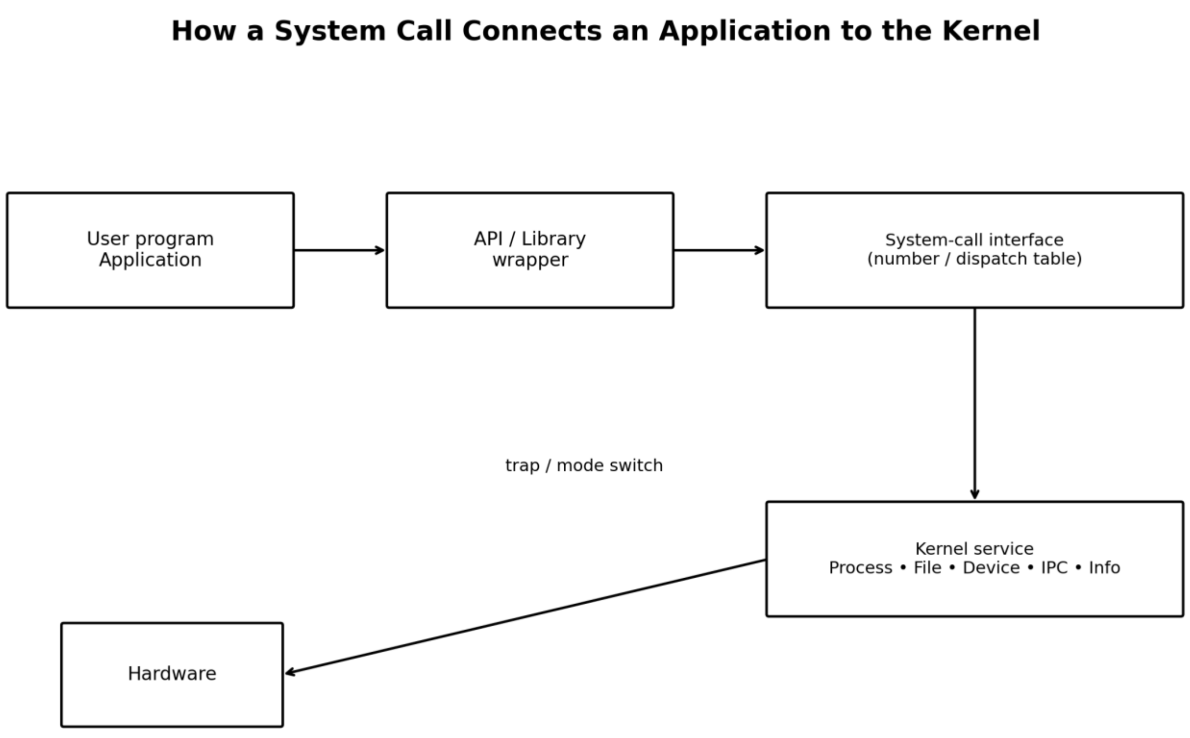
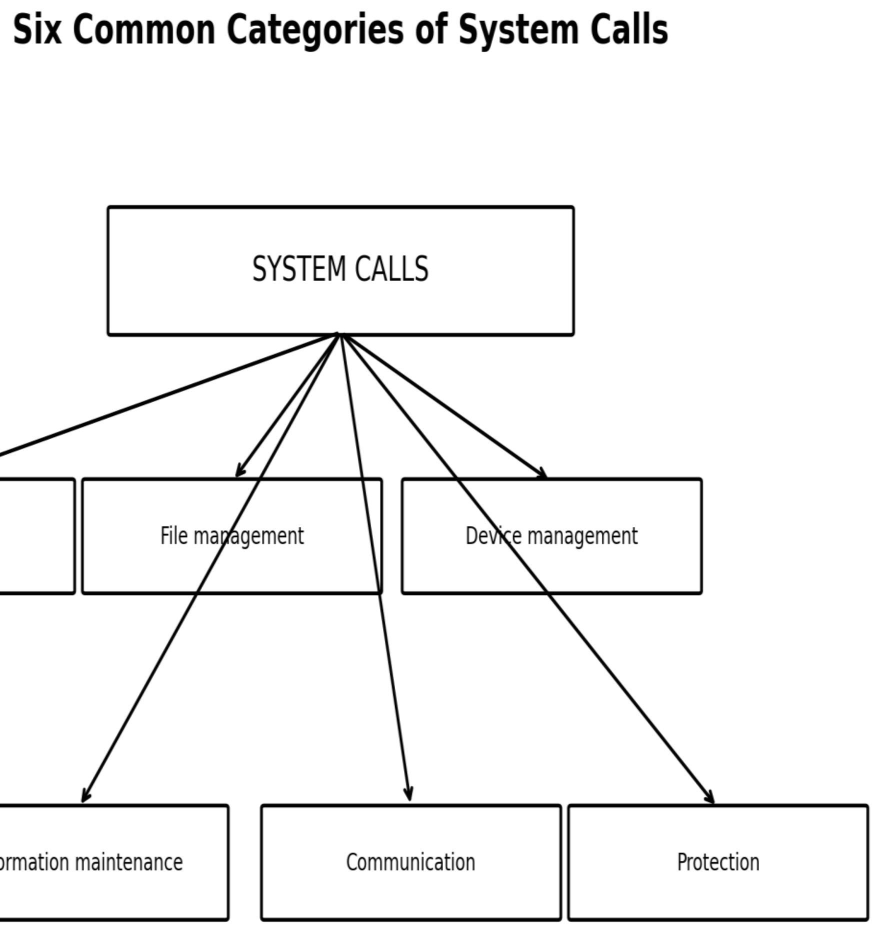


Figure 6. Simplified system-call path from application to kernel and hardware.

7.1 Why system calls are needed

- User programs operate with restricted privileges and should not directly control protected hardware resources.
- A system call provides a controlled transition from user mode to kernel mode.
- The OS validates the request, performs the privileged operation and returns a result/status.
- System calls form the interface between application software and OS services.

7.2 Six common categories of system calls



Category	Typical operations	Representative example
Process control	create/terminate processes, load/execute programs, wait and signal events	fork(), exec(), exit(), wait()
File management	create/delete, open/close, read/write, reposition, attributes	open(), close(), read(), write()
Device management	request/release devices, read/write, device attributes	device-specific I/O calls
Information maintenance	get/set time, system and process/file/device attributes	getpid(), time-related calls
Communication	create connections, send/receive messages, shared memory	pipe(), shmget(), shmat()
Protection	control access and permissions	access-control and permission-related calls

EXAM LINK | Summer 2026 Q3(a): list types of system calls with examples. Winter 2025 Q1(b): enlist four process-control system calls.

Silberschatz, Ch. 2 groups system calls into process control, file management, device management, information maintenance, communications and protection.

7.3 Open and close – a simple file-system example

- open(): requests access to a file and returns a file descriptor when successful in Unix-like systems.
- read()/write(): transfer data between the process and the opened file.
- close(): releases the file descriptor and associated kernel bookkeeping for that open instance.

EXAM LINK | The supplied notes include open() and close() as examples. These are useful illustrations of file-management system calls, although neither was directly asked in the two supplied papers.

8. Operating-System Components

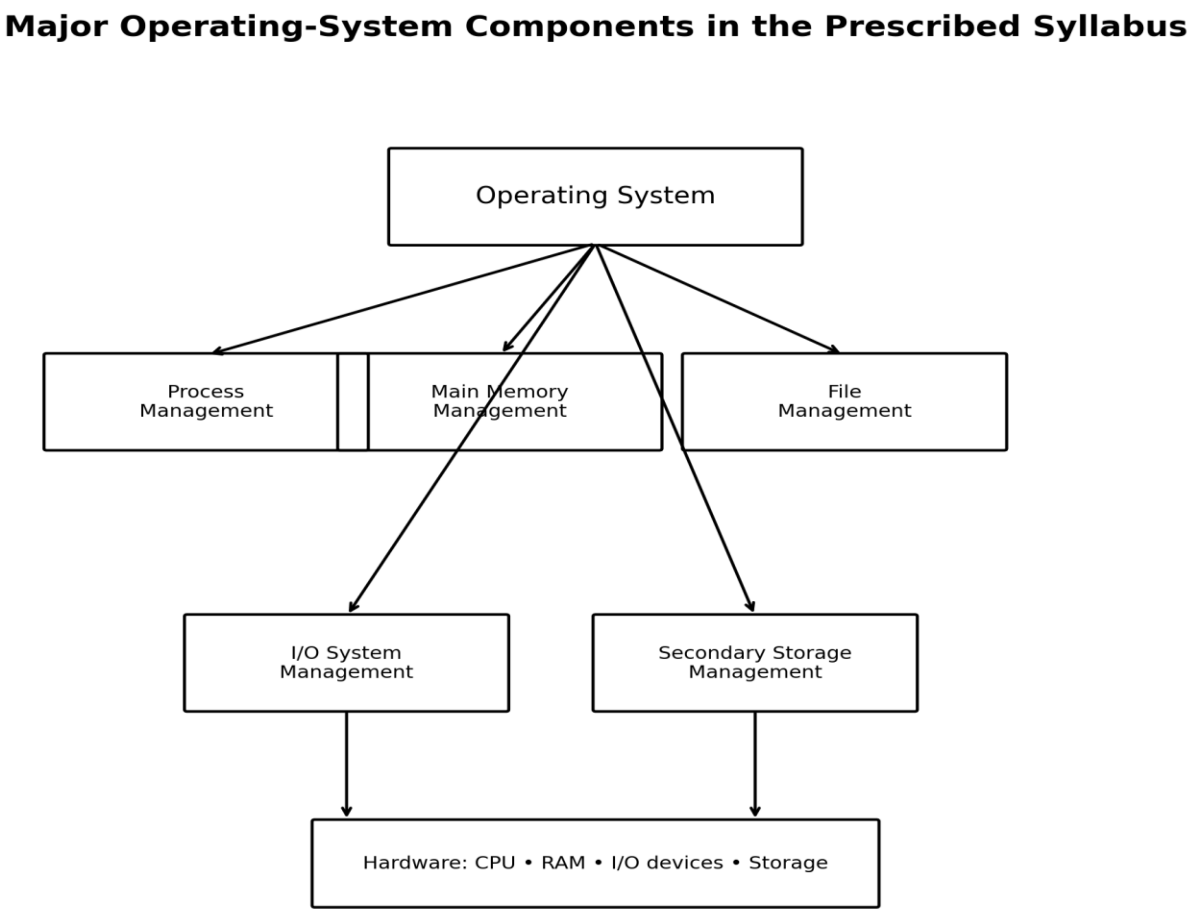


Figure 7. Five major components explicitly prescribed in Unit I.

Component	Major activities
Process management	Manages creation, scheduling, synchronization/communication support and termination of processes.
Main-memory management	Tracks which memory areas are in use, decides what is loaded, and allocates/deallocates memory.
File management	Creates/deletes files and directories, supports file manipulation, maps files to storage and supports backup.
I/O system management	Provides interfaces to I/O devices, manages device drivers and coordinates data transfer/completion.
Secondary-storage management	Manages free space, allocates storage and schedules storage-access requests.

EXAM LINK | Summer 2026 Q4(a): asks activities of main-memory management and process management. The full component list is prescribed under Unit I.

The syllabus explicitly lists Process Management, Main Memory Management, File Management, I/O Management and Secondary Storage Management as Unit-I components.

9. Content Review: What Was Moved / Added / Corrected

The supplied notes contain useful additional material on monolithic, layered and microkernel structures. These topics are not listed in the K-Scheme Unit-I content provided in the syllabus, and neither supplied question paper directly asks them. For a syllabus-controlled publication, they should be placed in an optional enrichment box or moved to an appendix rather than occupying core Unit-I pages.

Original notes issue	**Publishing decision**	**Reason**
Monolithic / layered / microkernel structures	Move to "Enrichment / beyond prescribed Unit I"	Not listed in the supplied Unit-I syllabus; not directly tested in the two supplied papers.
"OS is ATMA of computer" / "CPU is heart"	Remove from core notes	Informal and non-technical phrasing; does not support a precise learning outcome.
Linux/macOS shown as GUI-based only	Clarify	Both provide GUI and CLI environments; avoid an absolute classification.
Multitasking omitted from syllabus list but asked in Summer 2026	Retain as exam-linked extension	The actual question paper requires students to differentiate it from multiprogramming.
Mobile OS / Android was thin or absent	Add dedicated section	Explicitly prescribed in Unit I and asked in Winter 2025.
System-call categories lacked protection in some parts	Standardize to six categories	Matches standard textbook classification and improves exam coverage.

The supplied syllabus lists the prescribed Unit-I topics and separately identifies the recommended textbooks.

10. Quick Revision Sheet

Topic	**One-line recall**
OS	System software that manages resources and provides services to programs/users.
Batch OS	Jobs are collected and executed with little interactive intervention.

Multiprogramming	Keep multiple jobs in memory and switch when one waits, improving CPU utilization.
Multitasking	Multiple tasks make concurrent progress through rapid scheduling/context switching.
Time sharing	Interactive CPU sharing using short time slices/quanta.
Multiprocessor OS	Manages two or more cooperating processors.
Distributed OS/system	Coordinates autonomous networked computers and shared resources.
Real-time OS	Responds to events within timing constraints; hard vs soft real-time.
Mobile OS	OS optimized for mobile hardware, applications, touch, connectivity and power.
CLI	Text-command interaction.
GUI	Visual interaction using windows/icons/menus/pointers/touch.
System call	Controlled interface through which a program requests kernel services.
Five OS components	Process, main memory, file, I/O, secondary storage management.

Likely short-answer prompts from the supplied papers

- State any four operating-system services.
- Enlist four process-control system calls.
- What are the major differences between Linux and Windows?
- Differentiate multiprogramming and multitasking.
- List system-call types and give examples.
- Describe activities of main-memory management and process management.
- Explain real-time OS with applications.
- Explain Distributed OS and Mobile OS with examples.

11. Question-Paper Mapping: Unit-wise Questions

The following mapping classifies every sub-question in the two supplied papers against the prescribed five-unit syllabus. This is a content-alignment map, not a prediction of future questions.

Unit	Paper	Questions mapped	Frequency
Unit I	Summer 2026	Q2(a) Multiprogrammed vs multitasking; Q3(a) types of system calls; Q4(a) OS component activities; Q5(a) OS services.	4 sub-question opportunities
Unit II	Summer 2026	Q1(a) process states; Q1(b) ps/kill; Q2(b) schedulers; Q3(b) thread benefits; Q4(b) many-to-one model; Q6(a) IPC/shared memory.	6 opportunities
Unit III	Summer 2026	Q1(c) non-preemptive scheduling; Q1(d) CPU scheduler; Q2(c) Round Robin; Q4(c) scheduling criteria; Q5(b) deadlock prevention; Q6(c) FCFS/SJF turnaround.	6 opportunities
Unit IV	Summer 2026	Q1(e) fragmentation; Q1(f) virtual memory; Q3(c) paging vs segmentation; Q4(d) bitmap; Q5(c) paging hardware; Q6(b) FIFO/Optimal page faults.	6 opportunities
Unit V	Summer 2026	Q1(g) direct access; Q2(d) tree directory; Q3(d) file operations; Q4(e) contiguous allocation disadvantages.	4 opportunities
Unit I	Winter 2025	Q1(a) OS services; Q1(b) process-control system calls; Q1(d) Linux vs Windows; Q2(a) real-time OS; Q6(a) Distributed OS and Mobile OS.	5 opportunities
Unit II	Winter 2025	Q1(e) thread; Q3(a) process-state diagram; Q3(d) multithreading models; Q4(d) PCB; Q4(e) scheduler/type; Q5(b) Linux process commands.	6 opportunities
Unit III	Winter 2025	Q1(c) turnaround time/throughput; Q4(b) deadlock conditions; Q5(a) SJF/FCFS/Priority waiting time; Q5(c) deadlock/avoidance/prevention/Banker; Q6(b) SRTN/RR.	5 opportunities

Unit IV	Winter 2025	Q1(g) page fault; Q2(b) variable partitioning; Q2(c) bitmap/linked list; Q4(a) First/Best/Worst fit; Q4(c) paging vs segmentation; Q6(c) FIFO/Optimal page faults.	7 opportunities
Unit V	Winter 2025	Q1(f) file operations; Q2(d) two-level directory; Q3(b) file types/extensions; Q3(c) linked vs indexed allocation.	4 opportunities

Across the two papers, the largest number of sub-question opportunities were from Unit IV (13), followed by Unit II (12) and Unit III (11). Unit I and Unit V each had 9 and 8 opportunities respectively. This frequency should be treated as a past-paper observation, not a guarantee of future weightage.

EXAM LINK | For Unit-I publishing, every Unit-I question from both papers has been tagged above and cross-linked to the relevant sections in these notes.

12. References and Source Notes

[1] MSBTE, Operating System, Course Code 315319, Semester V, K Scheme, approved 24/02/2025. Unit I, TLO 1.1–1.4, CO1. Supplied syllabus copy.

[2] Abraham Silberschatz, Peter Baer Galvin & Greg Gagne, Operating System Concepts, 10th Global Edition, Wiley, ISBN 978-1-119-45408-3. Ch. 1–2 are especially relevant to OS role, services, interfaces and system calls. citeturn0search0

[3] William Stallings, Operating Systems: Internals and Design Principles, 9th Edition, Pearson. Relevant coverage includes OS components, system calls and process/scheduling concepts. citeturn1search1

[4] Dhananjay M. Dhamdhere, Operating System: A Concept-Based Approach, 3rd Edition, McGraw Hill Education, ISBN 978-1259005589. Listed in the supplied syllabus as suggested learning material.

[5] Richard Petersen, Linux: The Complete Reference, 6th Edition, McGraw Hill, ISBN 978-0071492478. Relevant to Linux shell, files, users, devices and system administration. citeturn1search2

[6] Richard Blum, Linux Command Line and Shell Scripting, Wiley India, ISBN 978-1118983843. Listed in the supplied syllabus.

[7] NPTEL, Operating System Fundamentals / Introduction to Operating System, IIT Kharagpur. The course covers introduction, OS structures, processes, threads, scheduling, virtual memory and file systems. citeturn0search1

Note on citations: textbook references are used as conceptual sources; the supplied syllabus itself names the recommended books. The external web sources were checked for bibliographic and topic-level support. The notes intentionally avoid reproducing long copyrighted passages and instead provide original, exam-oriented explanations.